PN - JP4332045 A 19921119
PD - 1992-11-19
PR - JP19910101467 19910507
OPD - 1991-05-07
TI - ARITHMETIC PROCESSOR
IN - SAITO YUTAKA
PA - FUJI ELECTRIC CO LTD
IC - G06F9/46

PN - JP4332045 A 19921119
PD - 1992-11-19
AP - JP19910101467 19910507
IN - SAITO YUTAKA
PA - FUJI ELECTRIC CO LTD
TI - ARITHMETIC PROCESSOR
AB - PURPOSE:To optionally set up the restarting position of an execution instruction after ending an interruption program by storing an instruction for changing the storage value of a stack register in the interruption program and executing the stored instruction by an instruction decoder and a writing circuit.
- CONSTITUTION:A program instruction read out from a program memory 20 is stored in a latch circuit 12 and decoded by an instruction decoder 11. When the instruction is a storage value changing instruction for the stack register 16, the decoder 11 generates an address change instruction detecting signal. At the time of inputting an interruption signal, a timing signal generating circuit 13 stores and outputs an interruption detection signal. At the time of inputting a write command signal, the circuit13 releases the storing state of the interruption detection signal. At the time of detecting an address change instruction in an interruption program, a writing circuit 15 writes an address value included in the address change instruction in the register 16.
I - G06F9/46

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（A）**　(11)特許出願公開番号

特開平4－332045

(43)公開日　平成４年(1992)11月19日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　9/46	３１０　Ｚ	8120－5B		

審査請求　未請求　請求項の数１(全　６　頁)

(21)出願番号　特願平3－101467

(22)出願日　平成３年(1991)５月７日

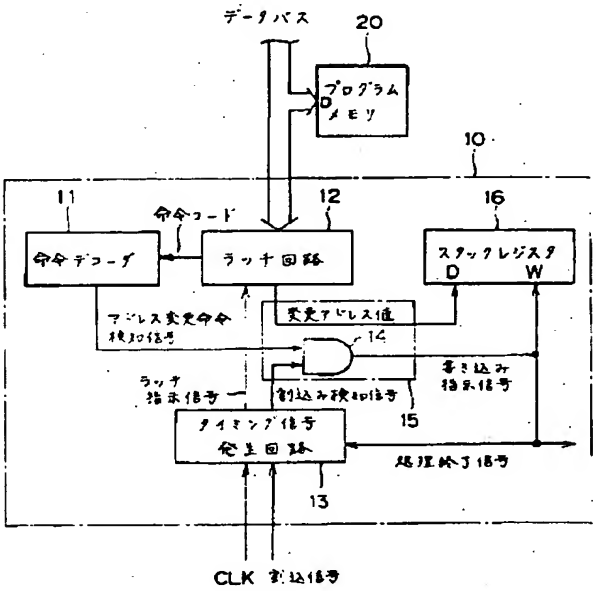
(71)出願人　000005234
富士電機株式会社
神奈川県川崎市川崎区田辺新田１番１号

(72)発明者　斉藤　豊
神奈川県川崎市川崎区田辺新田１番１号
富士電機株式会社内

(74)代理人　弁理士　谷　義一　(外１名)

(54)【発明の名称】　演算処理装置

(57)【要約】
【目的】　割込みプログラム終了後の実行命令の再開位置を任意に設定する。
【構成】　割込みプログラムの中にスタックレジスタ１６の格納値を変更する命令を記載し、この命令を命令デコーダ１１および書き込み回路１５で実行することにより実行再開のプログラム命令の位置を変更する。



1

【特許請求の範囲】

【請求項1】　第1のプログラムをプログラム命令単位で実行中に割込信号を入力した場合は、当該第1のプログラムにおいて実行しようとしていたプログラム命令の位置情報を記憶手段に記憶し、次に、割込用に定めた第2のプログラムを実行した後、前記記憶手段の位置情報の示すプログラム命令から実行を再開する演算処理装置において、前記記憶手段に記憶の前記位置情報を変更する旨を示す命令コードと、変更すべき新たなプログラム命令の位置情報とを含む変更命令を予め定め、実行対象のプログラム命令が前記変更命令であることを検出する命令識別手段と、該命令識別手段により前記変更命令が検出されたときには、当該変更命令に含まれる位置情報を前記記憶手段に更新的に書き込むことにより、前記第2プログラムの終了後に実行すべきプログラム命令を変更する変更手段とを具えたことを特徴とする演算処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、プログラムをプログラム命令単位で演算実行する演算処理装置に関する。

【0002】

【従来の技術】中央演算処理装置（CPU）は内蔵または外付のシステムプログラムメモリに格納された演算プログラムをプログラム命令単位で読出して、演算実行する。

【0003】通常、CPUはシステムプログラムメモリの開始アドレスをアドレスカウンタに設定して、先頭のプログラム命令を読出した後、プログラム命令の実行毎に、アドレスカウンタを“1”ずつ更新し、実行対象のプログラム命令を順次に読出す。

【0004】また、CPUの割込み端子に割込み信号を入力したときは、CPUは現在実行しているプログラムを一時中断し、割込信号に対応した別の割込み用プログラムを実行する。このプログラムを実行した後、CPUは中断していた元のプログラムの実行を再開することができるようになっている。割込み処理を実行する場合、CPU内の具体的なシステム処理は次のように行われる。

【0005】割込み信号を入力すると、CPUは、現在、アドレスカウンタに設定されているプログラム命令の読出しアドレスを割込みプログラム終了後の復帰先アドレスとして、スタックレジスタと呼ばれるCPU内の専用レジスタに格納する。

【0006】この後、CPUはアドレスカウンタに割込用プログラムの開始アドレスを設定し、アドレスカウンタの指示するシステムプログラム内のアドレスから割込み用プログラムを順次に読出し、実行する。割込み用プログラムの最後尾に記載された復帰命令（リターン命令）をCPUが読出すと、CPUはスタックレジスタに

2

格納されたアドレス値をアドレスカウンタに初期設定し、このアドレスカウンタの示すアドレスからプログラム命令の読出し演算実行を再開する。

【0007】このような割込処理を起動する割込み信号はCPUに対して複数点入力可能であり、複数の割込用プログラムを実行することが可能となっている。

【0008】

【発明が解決しようとする課題】しかしながら、従来この種、演算処理装置では割込み信号と割込用プログラムが1対1に対応しているので、多数の割込み処理を実行しなければならない場合は複数点の割込み信号を必要とする。

【0009】そこで、本発明の目的は、上述の点に鑑みて、割込み信号の点数に関係なく任意の本数の割込み用処理プログラムを実行させるのみならず、従来にはない新規の割込み処理を達成することの可能な演算処理装置を提供することにある。

【0010】

【課題を解決するための手段】このような目的を達成するために、本発明は、第1のプログラムをプログラム命令単位で実行中に割込信号を入力した場合は、当該第1のプログラムにおいて実行しようとしていたプログラム命令の位置情報を記憶手段に記憶し、次に、割込用に定めた第2のプログラムを実行した後、前記記憶手段の位置情報の示すプログラム命令から実行を再開する演算処理装置において、前記記憶手段に記憶の前記位置情報を変更する旨を示す命令コードと、変更すべき新たなプログラム命令の位置情報とを含む変更命令を予め定め、実行対象のプログラム命令が前記変更命令であることを検出する命令識別手段と、該命令識別手段により前記変更命令が検出されたときには、当該変更命令に含まれる位置情報を前記記憶手段に更新的に書き込むことにより、前記第2プログラムの終了後に実行すべきプログラム命令を変更する変更手段とを具えたことを特徴とする。

【0011】

【作用】本発明は、スタックレジスタと呼ばれる記憶手段に格納されたプログラム命令の開始位置に基づいて、割込み用第2プログラム終了後、再開すべきプログラム命令の位置が決定されることに着目し、割込み用第2プログラムの中に記載した変更命令を検出し、記憶手段の位置情報を変更することにより、たとえば他の割込みプログラムの先頭位置に直接の実行手順を移行させる。

【0012】

【実施例】以下、図面を参照して本発明実施例を詳細に説明する。

【0013】図1は本発明第1実施例の回路構成例を示す。

【0014】図1において、10は中央演算処理装置（CPU）を示す。

【0015】本実施例ではCPU10中の従来から周知

の回路についての説明を省略し、本発明に関わる回路構成のみ説明することにする。

【００１６】不図示のアドレスカウンタのアドレス指定によりプログラムメモリ２０から読出されたプログラム命令はデータバスに送出され、ラッチ回路１２に保持（ラッチ）される。ラッチ回路１２に対するラッチ指示信号はタイミング信号発生回路１３においてマシンクロックに基づき作成される。

【００１７】ラッチ回路１２に保持されたコード信号形態のプログラム命令は命令デコーダ（命令コード解読器）１１に入力され、命令デコーダにおいて命令内容の解読が行われる。

【００１８】命令デコーダ（本発明の命令識別手段）１１は従来の命令についての解読を行う他、読出しの命令コードが、本発明のスタックレジスタ１６に対する格納値変更命令（以下、アドレス変更命令と称す）を示す特定コードであることをコード比較により検出した場合、アドレス変更命令検知信号を発生する。

【００１９】タイミング信号発生回路１３はマシンクロックＣＬＫを入力し、ラッチ回路１２に対してラッチ指示信号を出力すると共に、割込信号を入力したときは割込み検知信号を保持出力する。また、タイミング信号発生回路１３は、後述の書き込み指示信号を入力したときは割込み検知信号の保持出力状態を解除する。

【００２０】書き込み回路１５は、割込みプログラム中のアドレス変更命令が検出されたときに、このアドレス変更命令中に含まれるアドレス値をスタックレジスタ（本発明の記憶手段）１６に書き込む。本実施例ではタイミング信号発生回路１３に保持された割込検知信号と命令デコーダ１１のアドレス変更命令検知信号が同時に発生したことをアンドゲート１４により検出し、スタックレジスタ１６に対する書き込み指示信号を発生する。

【００２１】このような回路構成において実行する割込み処理手順を図２のフローチャートを参照して説明する。図２のフローチャートは図１のプログラムメモリ２０に格納された制御手順を示す。この制御手順はメインプログラムと割込みプログラムから構成されている。メインプログラムはたとえば、一定周期で起動され、２１個のプログラム命令で構成されている。このメインプログラムはプログラムメモリ２０のアドレス“０”～アドレス“２０”（１０進数）に格納されているものとする。

【００２２】割込みプログラムは割込信号により起動され、４個のプログラム命令で構成されている。割込みプログラムはプログラムメモリ２０のアドレス“１００”～“１０３”（１０進数）に格納されているものとする。

【００２３】また、割込みプログラムの第２番目のプログラム命令（アドレス１０１）が本発明のアドレス変更命令となっている。このアドレス変更命令は命令の種類

を示す命令コードと、変更すべきアドレス値から構成される。本実施例では説明のためにメインプログラムの終了命令（図２参照）の格納されたアドレス値“２０”がアドレス変更命令の中に含まれているものとする。

【００２４】図１に戻り、回路動作を説明する。

【００２５】電源起動に応じて、ＣＰＵ１０内のアドレスカウンタにはプログラムメモリ２０の先頭アドレスと同じ値“０”が初期設定される。次にタイミング信号発生回路１３の発生する同期信号に従ってアドレスカウンタの計数値が不図示のアドレスバスを介してプログラムメモリ２０に転送され、プログラムメモリ２０のアドレス“０”からプログラム命令Ａ１（図２参照）が読出される。読出されたプログラム命令はデータバスを介してラッチ回路１２に保持される。

【００２６】命令デコーダ１１はラッチ回路１２に保持されたプログラム命令Ａ１の中の命令コード内容を解読し、その命令に対応する検知信号をオンする。この検知信号に応じてその命令を実行する論理回路、たとえばＣＰＵ内のレジスタの書き込み回路，読出し回路，演算回路等がその命令の示す情報処理を実行する。

【００２７】論理回路の処理終了信号に応じてアドレスカウンタが“１”に更新され、プログラムメモリ２０のアドレス“１”から、次ステップのプログラム命令Ａ２が読出され、ラッチ回路１２に保持された後、上述と同様の処理手順でプログラム命令Ａ２の示す情報処理が実行される。また、タイミング信号発生回路１３はオンの割込み検知信号をアンドゲート１４に保持出力する。

【００２８】このようにして、アドレス“１０”のプログラム命令Ａ３を実行しようとするときに割込み信号がＣＰＵ１０に入力されると、従来周知の処理手順により、アドレスカウンタの計数値“１０”がスタックレジスタ１６に格納され、次にＣＰＵ１０内のレジスタに格納された、割込プログラムの先頭アドレス値“１００”がアドレスカウンタに設定される。

【００２９】この結果、ＣＰＵ１０の実行命令として図２に示すように、プログラムメモリ２０のアドレス“１００”からプログラム命令Ｂ１が読出される。

【００３０】このプログラム命令の実行終了の後、アドレスカウンタが“１０１”に更新されると、アドレス変更命令がプログラムメモリ２０から読出される。

【００３１】ラッチ回路１２に保持されたこのアドレス変更命令が命令デコーダ１１により解読されると、アドレス変更命令検知信号がオンとなる。この結果、アンドゲート１４の２入力が共にオンとなるのでゲートが開き、書き込み指示信号がオンレベルとなる。

【００３２】スタックレジスタ１６はオンの書き込み指示信号により、ラッチ回路１２のプログラム命令中の変更アドレス値を上書き記憶することによりアドレス変更を行う。

【００３３】以下、ＣＰＵ１０はアドレスカウンタを実

5

行して順次に割込プログラムのプログラム命令を実行する。

【００３４】割込プログラムの最後尾に位置する復帰命令をＣＰＵ１０が読出し、命令デコーダ１１によりその命令内を解読すると、従来処理同様、アドレスカウンタの設定値がスタックレジスタ１６の格納値“２０”に更新され、プログラムメモリ２０のアドレス“２０”から終了命令（図２参照）が読出される。この結果、ＣＰＵ１０はメインプログラムを終了し、命令待機状態となる。

【００３５】以上、説明した例は、割込み信号によりメインプログラムを中断し、割込みプログラムを実行した後、メインプログラムを終了してしまう例であるが、アドレス変更命令の変更値を変えるのみで種々のプログラムを実行することができる。この実行形態を図３，図４に示す。

【００３６】図３は、第１プログラムの実行中に割込信号を受けた場合、割込プログラムの終了後、第２プログラムを先頭プログラム命令から実行する例および割込プログラムの終了後、第３プログラムの途中のプログラム命令から実行する例を示す。

【００３７】図４はループ形態で構成された第４プログラムから割込み信号により脱却し、割込プログラムを終了後別の第５プログラムにＣＰＵ１０の実行手順を移行させる例を示す。

【００３８】このような形態でプログラムを実行させることは復帰命令で実行手順を元に戻してしまう従来例ではできなかったことである。

【００３９】以上、説明した第１実施例はプログラムメオリにリードオンリメモリ（ＲＯＭ）を用い、プログラム命令がマシン語と呼ばれるＣＰＵを作動させるプログラム言語で記載された演算処理装置に好適な例である。次に、実行対象のプログラムがベーシック，フォートラン，その他高級言語や中間言語と呼ばれる言語で登録記憶される演算処理装置に好適な第２実施例を図５を用いて説明する。

【００４０】図５において、ＣＰＵ１００，システムプログラムメモリ１２０，アプリケーションプログラムメモリ１３０，ワークメモリ１４０がバスに共通接続されている。

【００４１】ＣＰＵ１００はマシン語形態のプログラム命令を実行するＣＰＵチップであればよく、従来から知られているものを使用することができる。

【００４２】システムプログラムメモリ１２０には装置制御たとえば、バス接続されたメモリに対して情報を読み／書きする処理を実行するためのシステムプログラムと、中間言語のプログラム命令をマシン語のプログラム命令に翻訳するためのインタプリタと呼ばれるプログラムとが格納されている。

【００４３】アプリケーションプログラムメモリ１３０

6

はユーザの作成した中間言語形態の演算対象の（アプリケーション）プログラムを格納する。アプリケーションプログラムは不図示のフロッピーディスク等からこのメモリにＣＰＵ１００により転送される。

【００４４】ワークメモリ１４０はＣＰＵ１００の演算やシステム処理に用いるデータを格納する。本実施例では割込みプログラム終了後のＣＰＵ１００の実行処理の戻り先を示すアドレス情報（スタックポインタと称す）はワークメモリ１４０内の専用領域に格納する。なおＣＰＵ１００内のスタックレジスタは、システムプログラムを読出す際にアプリケーションプログラムの読出しアドレスを退避記憶させるために用いる。

【００４５】このような構成において、ＣＰＵ１００はアプリケーションプログラムの先頭アドレスから中間言語のプログラム命令を読出すと、ワークメモリ１４０に一時記憶し、次に、インタプリタ１２１に従って、マシン語形態の命令に変換した後、変換後の命令を実行する。このとき、接続のメモリ１４０等に対して情報の読み／書きを行う場合は、システムプログラム１２０を用いて命令に規定された内容を実行する。このような処理はインタプリタ方式の演算処理と呼ばれ、周知であるので詳細な説明は要しないであろう。

【００４６】本実施例では、インタプリタ１２１の中に、スタックポインタの変更命令検出処理（図６のステップＳ１００）およびスタックポインタの変更処理（図６のステップＳ１１０）を実行させるプログラム手順を記載しておく。

【００４７】変更命令検出処理はより具体的にはアプリケーションプログラムメモリ１３０から読出した実行対象のプログラム命令（中間語）がスタックポインタの変更命令であるか否かをコード比較により判定する処理である。

【００４８】スタックポインタ変更処理は、アプリケーションプログラムメモリ１３０から読出した上記スタックポインタの変更命令の中から変更値を抽出し、次に、ワークメモリ１３０のスタックポインタに変更値を書き込む処理をマシン語に翻訳する処理である。

【００４９】このようなインタプリタを用いることによって、アプリケーションプログラム中の割込みプログラムに記載したスタックポインタの変更命令が検出されると、スタックポインタの変更処理内容がマシン語に翻訳され、ＣＰＵ１００により実行される。

【００５０】この結果、アプリケーションプログラム中の割込プログラムを終了した後のＣＰＵ１００の実行手順の戻り先はスタックポインタの変更値により決定される。

【００５１】第１実施例はハードウエアにより本発明を実現する例であり、第２実施例はソフトウエアにより本発明を実現する例であるが、演算処理装置を用いる電子機器の仕様に応じて所望の形態を採用するとよい。

7

【００５２】本実施例の他に、次の例が挙げられる。

【００５３】１）本発明実施例ではスタックポインタの変更命令を単に検出するようにインタプリタを構成しているが、検出の変更命令が割込みプログラム中にあることを確認してスタックポインタの変更を行うこともできる。この場合、割込プログラムを実行する際にＣＰＵ内の特定レジスタに割込プログラムを実行する旨のフラグ情報を設定し、上記変更命令を検出した際にこのフラグ情報を参照する。

【００５４】２）第１実施例のスタックレジスタまたは第２実施例のスタックポインタに格納するアドレス値は１つにしているが、割込み処理をネスト的（１つの割込み処理を実行中に新たな別の割込み処理を実行すること）に実行する演算処理装置ではスタックメモリに累積的に、戻り先情報を格納する。このような演算処理装置では、最後に書き込まれた戻り先情報を変更命令により変更するとよい。

【００５５】また、プログラムの実行手順によっては、変更情報のみを残し、他の戻り先情報を消去することも考えられる。

【００５６】３）本発明実施例ではスタックメモリの情報を書き換えるために変更命令を新たに設けているが、スタックメモリのアドレスが固定化されていて誤指示の発生を容認する場合は通常の書き込み命令で書き込み処理を実行することもできる。

【００５７】

【発明の効果】以上、説明したように、本発明では、割

8

込プログラムの終了後の移行手順を任意所望のプログラム命令位置に設定できるので、たとえば２つの割込プログラムを連続的に実行したり、ループ処理手順を割込み処理により脱却させるという従来にはない新たな演算処理を実行することができる。

【図面の簡単な説明】

【図１】本発明第１実施例の回路構成を示すブロック図である。

【図２】本発明第１実施例のＣＰＵ１０のプログラム命令処理手順を示すフローチャートである。

【図３】本発明第１実施例の他のプログラム命令実行手順を示すフローチャートである。

【図４】本発明第１実施例の他のプログラム命令実行手順を示すフローチャートである。

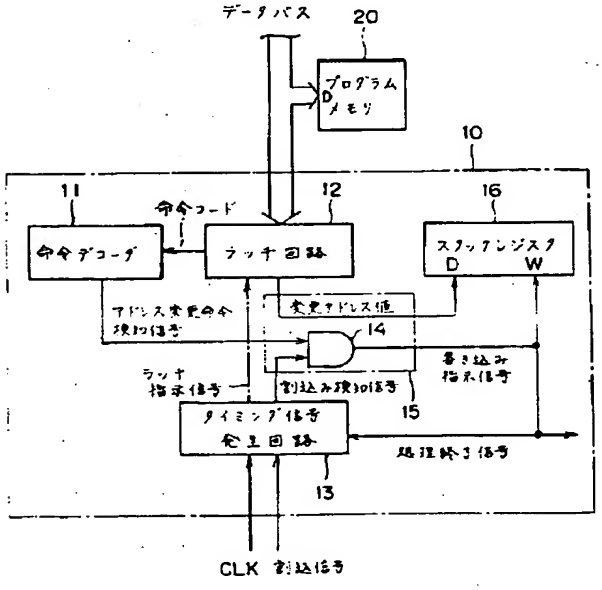
【図５】本発明第２実施例の回路構成を示すブロック図である。

【図６】図５のインタプリタ１２１の処理内容を示すフローチャートである。

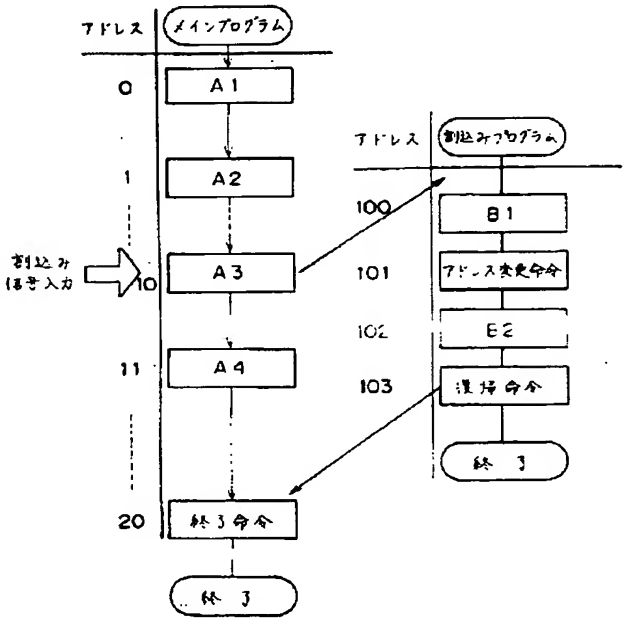
【符号の説明】

１０　ＣＰＵ
１１　命令デコーダ
１２　ラッチ回路
１３　タイミング信号発生回路
１４　アンドゲート
１５　書き込み回路
１６　スタックレジスタ
２０　プログラムメモリ

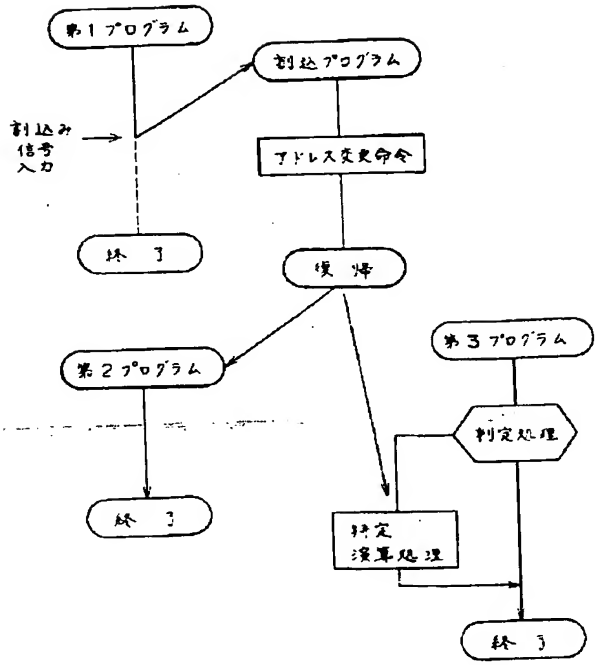
【図１】



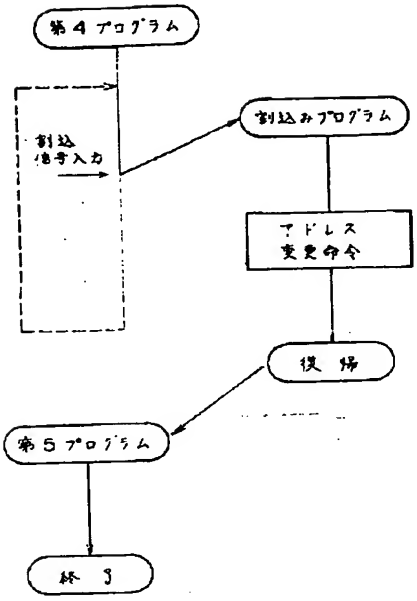
【図２】



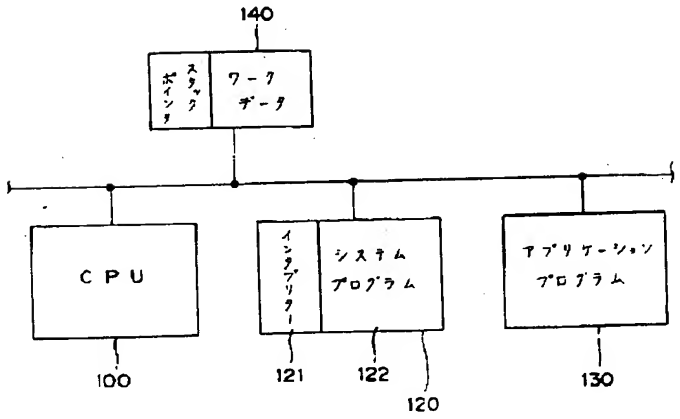
【図3】


第1プログラム
割込プログラム
割込み信号入力
アドレス変更命令
終了
復帰
第2プログラム
第3プログラム
判定処理
終了
判定演算処理
終了

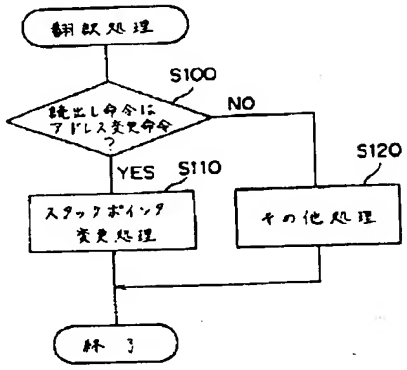
【図4】


第4プログラム
割込信号入力
割込みプログラム
アドレス変更命令
復帰
第5プログラム
終了

【図5】


140
スタックポインタ
ワークデータ
CPU
インタプリター
システムプログラム
アプリケーションプログラム
100
121
122
120
130

【図6】


翻訳処理
S100
読出し命令はアドレス変更命令か？
NO
YES
S110
S120
スタックポインタ変更処理
その他処理
終了